

# COMP2ProjetM1

## Compilateur DOMUS pour simulateur SiDo

*Le lann Addwyn  
Ledoussal Daniel*

| Travail réaliser       | État                 | Choix de conception  |
|------------------------|----------------------|--|
| Analyse lexicale       | Réalisé complètement | On a décidé de regrouper les attributs dans un seul token. De même pour les méthodes. Nous avons séparé les mots clés pour faciliter l'implémentation. |
| Analyse syntaxique     | Réalisé complètement | Nous avons réalisé la grammaire en nous inspirant des méthodes vu en TD.   |
| Production du code     | Réalisé complètement |  |
| Analyse sémantique     | Réalisé complètement | Nous avons utilisé des tables de symboles pour vérifier la bonne déclarations des variables.   |
| Génération d'un résumé | Réalisé complètement |  |

Grammaire CUP :

```

prog → PROG_DOMUS_DEBUT bloc_appareil bloc_interface bloc_scenarii
      bloc_commande PROG_DOMUS_FIN
bloc_appareil → DECLA_APPAREIL_DEBUT list_appareil DECLA_APPAREIL_FIN
list_appareil → list_variable_appareil list_appareil | list_def_appareil
list_appareil | ε
list_variable_appareil → TYPE nom_variable_appareil POINT
nom_variable_appareil → VARIABLE VIRGULE nom_variable_appareil | VARIABLE |
error
list_def_appareil → DEFINIR VARIABLE EGAL ACC_OUVERT nom_variable_appareil
ACC_FERMER POINT
bloc_interface → DECLA_INTERFACE_DEBUT list_interface DECLA_INTERFACE_FIN
list_interface → list_variable_interface list_interface | ε
list_variable_interface → TYPE nom_variable_interface POINT
nom_variable_interface → VARIABLE VIRGULE nom_variable_interface | VARIABLE |
error
bloc_scenarii → DECLA_SCENARII_DEBUT list_bloc_scenario DECLA_SCENARII_FIN
list_bloc_scenario → bloc_scenario list_bloc_scenario | ε
bloc_scenario → DECLA_SCENARIO_DEBUT VARIABLE SUPERIEUR list_instruction
DECLA_SCENARIO_FIN VARIABLE SUPERIEUR | error

```

```

list_instruction → list_variable_scenario list_instruction | ε
list_variable_scenario → MESSAGE PAR_OUVERT list_parametre_message PAR_FERMER
POINT_VIRGULE | SI PAR_OUVERT si_parametre type_comparaison si_parametre
PAR_FERMER ALORS list_instruction sinon_comparaison | VARIABLE POINT METHODE
POINT_VIRGULE | EXECUTER VARIABLE POINT_VIRGULE | POURTOUT VARIABLE DEUX_POINT
pourtout_list FAIRE POINT_VIRGULE
si_parametre → VARIABLE POINT METHODE | ATTRIBUT
list_parametre_message → parametre_message list_parametre_message | ε
parametre_message → VARIABLE suite_variable_message | CHAINE
suite_strOuMeth_message
suite_variable_message → POINT METHODE suite_strOuMeth_message | VIRGULE | ε
suite_strOuMeth_message → VIRGULE | ε
pourtout_list → TYPE | VARIABLE
type_comparaison → DOUBLE_EGAL | DIFFERENT
sinon_comparaison → SINON list_instruction FSI POINT_VIRGULE | FSI
POINT_VIRGULE
bloc_commande → DECLA_COMMANDE_DEBUT list_commande DECLA_COMMANDE_FIN
list_commande → commande list_commande | ε
commande → motclef_commande VARIABLE EGAL diff_commande POINT
motclef_commande → ASSOCIER | PROGRAMMER
diff_commande → VARIABLE | dateType | ACC_OUVERT list_dateType ACC_FERMER |
ACC_OUVERT list_variable_commande ACC_FERMER
dateType → PAR_OUVERT dateParametre VIRGULE dateParametre VIRGULE dateParametre
VIRGULE dateParametre VIRGULE dateParametre PAR_FERMER
dateParametre → ENTIER | UNDERSCORE
list_dateType → dateType VIRGULE list_dateType | dateType
list_variable_commande → VARIABLE VIRGULE list_variable_commande | VARIABLE

```